HAL open science

You'll Never Tweet Alone Building Sports Match Timelines from Microblog Posts

Amosse Edouard, Elena Cabrio, Nhan Le Thanh, Sara Tonelli

To cite this version:

Amosse Edouard, Elena Cabrio, Nhan Le Thanh, Sara Tonelli. You'll Never Tweet Alone Building Sports Match Timelines from Microblog Posts. RANLP17, Jul 2017, Varna, Bulgaria. hal-01561441

HAL Id: hal-01561441
https://inria.hal.science/hal-01561441

Submitted on 19 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

You'll Never Tweet Alone
Building Sports Match Timelines from Microblog Posts

Amosse Edouard
Inria, CNRS, I3S
Nice, France
edouard@unice.fr

Elena Cabrio
Inria, CNRS, I3S
Nice, France
cabrio@unice.fr

Sara Tonelli
Fondazione Bruno Kessler
Trento, Italy
satonelli@fbk.eu

Nhan Le-Than
Inria, CNRS, I3S
Nice, France
le-than@unice.fr

Abstract

In this paper, we propose an approach to build a timeline with actions in a sports game based on tweets. We combine information provided by external knowledge bases to enrich the content of the tweets, and apply graph theory to model relations between actions and participants in a game. We demonstrate the validity of our approach using tweets collected during the EURO 2016 Championship and evaluate the output against live summaries produced by sports channels.

1 Introduction

Historically, sports fans have watched matches either at the stadium or on TV, or have listened to them on the radio. In the latest years, however, social media platforms, in particular microblogs, have become a new communication channel also to share information and comment on sports events, thus creating online communities of sports fans around the world. Microblogs are particularly suitable for this, thanks to their coverage and speed, making them a successful channel to follow and comment on events in real time. Also sports teams and medias have benefited from these platforms, using them to extend their contact networks, increase their popularity and exchange information with fans (Gibbs and Haynes, 2013; Özsoy, 2011). The need to monitor, categorize and organize information about the matches is particularly relevant during large events like the Olympic Games or FIFA World Cup: several matches take place in a limited time span, sometimes in parallel, and summaries are manually made by journalists who take notes of the main actions during the matches. A few approaches have recently tried to perform this task automatically by recognizing actions in multimedia data such as videos, transcripts of matches or news (Hannon et al., 2011; Snoek et al., 2003; Snoek and Worring, 2005).

In this work, we investigate whether the same task can be performed relying only on user-generated content from microblogs. In fact, opinions shared by fans during sports matches are usually reactions to what is happening in the game, implicitly conveying information on the ongoing events. Existing works aimed at building complete summaries of sports games from tweets (Nichols et al., 2012; Xu et al., 2013) used simple approaches based on the observation of peaks in the tweets' volume. Even though such approaches effectively detect the most salient actions in games (e.g. goals), they fail to capture actions that are not reported by many users (e.g. shoots). Moreover, they focus only on specific information related to the events in sports games. For example, (Löchtefeld et al., 2015) are interested in detecting only goals, yellow and red cards in soccer games, ignoring the players involved in the actions, while (Alonso and Shiells, 2013) only detect time and keywords describing sub-events, ignoring the players that are involved.

In this paper we perform a more complex task: we create a fine-grained, real-time summary of the sub-events occurring in sports games using tweets. We define a sub-event in a match as an action that involves one or many participants (e.g. a player, a team) at a given time, as proposed by (Dou et al., 2012). More specifically, we want to address the following research questions:

- Is it possible to build detailed sports games summaries in a unsupervised fashion, relying only on a controlled vocabulary?
- To what extent can Twitter be used to build a complete timeline of a game? Is information retrieved via Twitter reliable and sufficient?

The paper is organized as follows. Section 2 reviews existing literature on the topic; Section 3 presents the approach we propose, and Section 4 outlines the experimental setting and the obtained results. Conclusions end the paper.

2 Related Work

Twitter as a source of data has gained tremendous attention in several research fields such as Information Retrieval and Natural Language Processing (NLP). In the latest years, there has been a bulk of work on event detection and event tracking: this section discusses works that analyze the content of tweets for tracking major events, and more specifically sports events.

Most of the approaches to track sports events are based on spike detection on the stream of messages, in order to detect sub-events. To summarize event streams, (Nichols et al., 2012) propose a method that identifies spikes in Twitter feed and selects tweets from a sub-event by scoring each of them based on phrase graph (Sharifi et al., 2010). This method may produce unexpected summary if most of the tweets published during the spike are not related to the sub-event. In (Kubo et al., 2013) live sports summary are generating by prioritizing tweets published by good reporters (defined a users who posts informative tweets right after an important event has occurred in the event stream of an identified event). First, they identify spikes in the stream of an event as indicators of sub-events, and then the system tries to generate a summary by measuring the explanatory of the tweet by the presence of player's names, team names and terms related to the event. Similarly, in (Alonso and Shiells, 2013) when a spike is detected, the tweets published during the period are analyzed to identify the most frequent terms which they use to describe spikes in a tweets' histograms (spikes are considered as sub-events).

To summarize tweets related to football games, (Jai-Andaloussi et al., 2015) create event clusters with similar documents (according to cosine similarity), that are then automatically classified as relevant to football actions. This method requires training data for cluster classification.

In the peculiar case of sports games, spikes do not necessarily characterize a sub-event. For example, when the crowd disagrees with the referees or a player, emotional tweets to express disagreement are published. On the other hand, actions with low importance (e.g. a shoot) or actions produced by non-popular teams or players (e.g. Albania) may not produce peaks in the volume of tweets. Thus, approaches solely based on spikes detection will be unable to capture those actions. In our approach, we rely on Named Entities (NEs) to identify whether or not a tweet is related to a sports event. Besides, we rely on an adaptive threshold tuned according to the actions and the team (or player) of interest to evaluate whether or not the actions should be added to the timeline.

3 Proposed Approach

This section describes the approach we propose to detect sub-events in sports games and to build a timeline (Figure 1). Although the approach is general-purpose, we take as an example soccer games, so that we can use a consistent terminology (e.g. teams, penalties, players, etc.). The pipeline can be applied to any sports as long as it is represented in the Sports Markup Language.

First, a module for information extraction identifies actions (e.g. goals, penalties) and participants (e.g. player's names, teams) mentioned in tweets, setting relations between them (see examples in Table 1). Then, participants, actions and relations are modeled together in a temporal event-graph, taking into account also the time of the tweet. This leads to the creation of a timeline where actions and participants are connected and temporally ordered. The modules of this pipeline are described in detail in the following Sections.

Tweets	Action	Particip.
kick off.... #engwal #euro2016 #teamengland	D1P	england wales
how has ramsey not got a yellow card yet every attempt to tackle has been a foul.	CJA	ramsey wales
goaaaaaaaaaaal from bale woah #eng 0-1 #wal	BUT	bale wales

Table 1: Example of input tweets and detected actions and participants in the game played on June 16, 2016 between England and Wales. D1P: First period begins, CJA: Yellow card, BUT: Goal.

3.1 Information Extraction

The first module of the timeline extraction pipeline retrieves participants and sub-events (or actions)[1] from tweets, and sets relations between them. In

[1]In this paper we use interchangeably the terms actions and sub-events to refer to actions in a sports game.

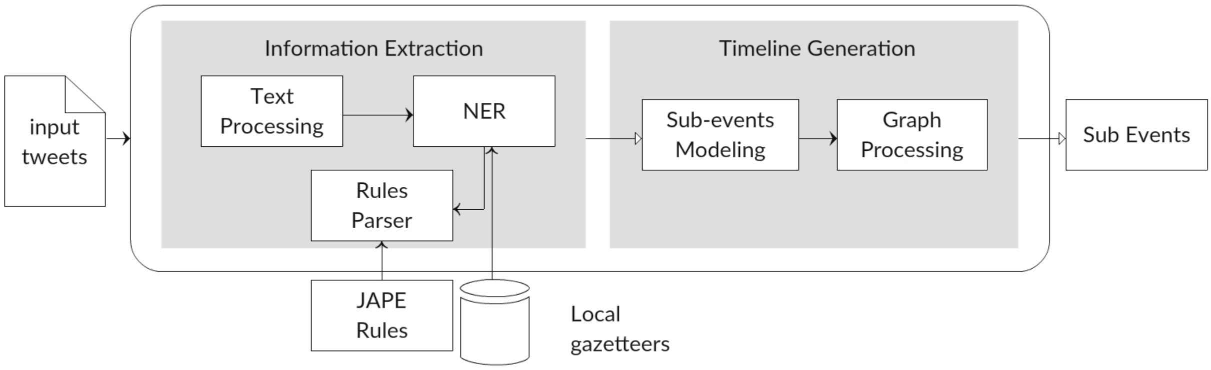

Figure 1: Sub-events extraction pipeline in which data is flowing in the sense of the arrows. The output are the sub-events detected from the input tweets.

the case of soccer, actions are defined by FIFA (Fédération Internationale de Football Association), e.g. goals, penalties, yellow/red cards, etc. Participants are the actors who induce the actions. For soccer games, they are players and teams.

For the information extraction task we use GATE (Cunningham et al., 2002), because it includes a highly flexible Named Entity Recognition (NER) tool that allows the integration of custom gazetteers. Indeed, in order to detect *actions*, we update its gazetteer based on the Sports Markup Language (Council, 2017), a controlled vocabulary used to describe sports events. SportsML core schema provides concepts allowing the description of events for 11 major sports including Soccer, American football, Basketball and Tennis. For soccer games, we extract actions such as goals, substitutions, yellow/red cards and penalties. Furthermore, we enrich the list of actions with synonyms extracted from Wordnet (Fellbaum, 1998).

As for *participants*, we update the gazetteer using the football-data API[2] that, given a soccer game in input, returns the name of the teams and their players. We also apply some heuristics so as to associate different spelling variations to players' and teams' names. This is done by considering separately or by combining the different parts of the players' names (i.e. first name and lastname). For instance, *"giroud"*, *"oliviergiroud"* or *"olivier_giroud"* are all associated with *"Olivier Giroud"*, a player in the French national team.

When launching GATE, we first pre-process the data using the in-built tweet normalizer, tokenizer and PoS-tagger. Then, we use the NER module including the two custom gazetteers we created as described before. We also set links representing relations between actions and participants by means of JAPE (Java Annotation Pattern Engine) rules, a GATE-specific format to define regular expressions needed for pattern matching. Since relations detected through JAPE rules tend to be very accurate, we assign a weight $= 2$ to edges extracted from such rules. If an action and a participant appear in the same tweet but are not matched through a JAPE rule, we set a link with a lower weight $= 1$, to account for a lower precision.

[2] http://api.football-data.org

3.2 Timeline creation

This section describes how we build a timeline describing a match from the extracted list of actions, participants and their relationships.

Modeling sub-events. The output of the information extraction module (Figure 1) is a list of tuples $\langle a, p, t, \omega \rangle$, where a is a sports action, t the timestamp of the tweet and p the participant involved and ω is the weight of the edge connecting a and p. These tuples are used to build a temporal event graph (see Figure 2). To retain temporal information on the sub-events, we split the game in fixed time windows, and create an event-graph that models the relationships between actions and participants for each time window. We refer to such graphs as *temporal graphs* (Verhagen et al., 2007) and we build them as follows:

- *Nodes*: Actions and participants are represented by nodes in the event-graph. First, we retrieve the nodes of the actions, and then we add the connected participants nodes;
- *Edges*: Nodes are connected by an edge if a relation can be set in the tweets published

during the time-window. The occurrence of this relation is used to increase the weight of the edges. Relationships between participants are created for actions involving 2 or more participants (e.g. a substitution).

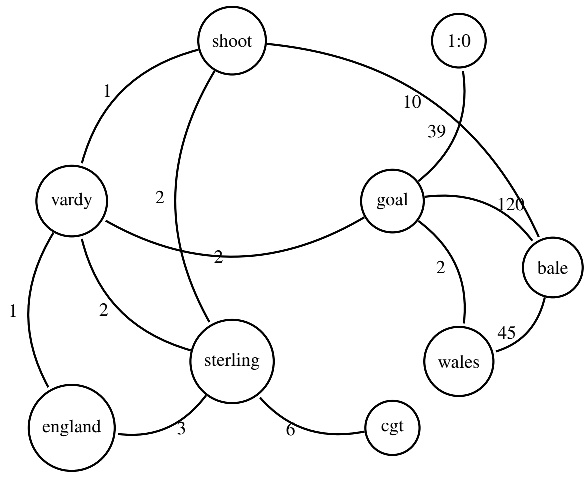

Figure 2: Example of the event-graph for the game between England and Wales at time-window 22.

Figure 2 shows a temporal graph at time-window 22 of the game between England and Wales (Game #16 on June 16, 2016). In this example, we observe edges linking participants, e.g. connecting the node "Sterling" and "Vardy", retrieved from tweets requesting the substitution of "Sterling" by "Vardy". These are both linked also to the node "England', i.e. their team.

Processing the Event-Graphs. At this stage, the weighted relations between actions and participants are considered as sub-event candidates. We cannot automatically include them in the timeline because they could represent opinions or wishes of the fans: when the supporters disagree with a call by the referees, they usually express their disagreement by tweeting the actions that should have been called. For example, users may ask for penalties or a yellow card after a fault by a player, as in the following tweet: *"how has ramsey not got a yellow card yet every attempt to tackle has been a foul"*. In general terms, we may assume that real sub-events in a game are reported by many users, while, on the contrary, an action reported only by a few users is more likely to be a subjective post reflecting a user's opinion (for example, s/he thinks that a player could have done a better choice).

In most of the existing work, an empirical threshold is set to measure the importance of the actions reported in tweets (Alonso and Shiells, 2013; Marcus et al., 2011). However, we observe that the number of tweets generated for a given action is highly dependant on the game and the team or player involved. For instance, the number of tweets reporting the goal scored by Romania against France (match #1: June 10, 2016) was twice lower than the number of tweets reporting a shoot by Rooney in the beginning of the match between England and Wales. Thus, we find it useful to tune the thresholds by taking into account both the type of the action and the popularity of the teams involved in the game.

For each action belonging to a certain sport, we manually define an empirical threshold according to the importance of the action. For soccer, we can assume that a goal will trigger a higher number of tweets than a shoot. These empirical values can be defined by domain experts for each category of the sports we want to track. Based on the predefined thresholds, the interest of the games for people and the popularity of the opponent teams, we adjust the empirical thresholds using Kreyszig standard score formula (Kreyszig, 2007) as follows :

$$\varphi_{a,t} = \epsilon_a * \frac{\eta_{g,t} - \overline{\eta}_g}{\sigma_g} \qquad (1)$$

where $\varphi_{a,t}$ is the threshold for action a at time t of the game, ϵ_a the empirical threshold for a, $\eta_{g,t}$ the count of tweets related to the game at time t, $\overline{\eta_g}$ the mean count, σ_g the standard deviation of tweets related to the game in the past time windows.

Ranking Sports Actions. Let $A = \langle a, p, t, \omega \rangle$ be a quadruplet modeling an action a at time t, involving participants p and weighted by ω (i.e. the number of edges connecting a and p in the event graph). For each participant, we compute a standard score as follows:

$$z_{a,p,t} = \frac{\eta_{\omega_i} - \overline{\eta}_{\omega_i}}{\sigma_{\omega_i}} \qquad (2)$$

where η_ω is the weight of the edge in graph G that connects nodes a and p, $\overline{\eta}_\omega$ is the mean count of all the actions of type a induced by p, and σ_ω is the standard deviation of relationship between a and p over all past time windows. Thus, we evaluate the action by taking the ratio between the standard score for each participant and the total standard scores for all the participants as follows :

$$z_{a,t} = \frac{z_{a,p_i,t}}{\sum\limits_{p_i \in P} z_{a,p_i,t}} \qquad (3)$$

At a given time t an action is added to the timeline iff there exists at least a participant p such that $z_{a,t} \geq \varphi_{a,t}$.

As shown in Algorithm 1, we first merge the current event graph and the graph from the previous time window (Line 1). Then, from the merged graph, we collect all vertices of type $foot\_action$ and for each we retrieve all connected nodes as participants of the action (Lines 4-6). We compute the adaptive threshold for each action and a standard score for each participant using equation 1 and 2, respectively (Lines 7-9). Finally, sub-event candidates are created with participants that have a score higher than the threshold of the action (Lines 10-16). It is important to notice that, for some actions, participants may not be required (e.g. beginning/end of periods in soccer), for such actions we consider both teams as participants in order to comply with equations (2 and 3). We remove from the event graph actions and participants involved in sub-events. Besides, nodes that were not related to sub-events are kept to be processed in the next time-window. However, if a node cannot be confirmed as related to sub-events in two consecutive time windows, we consider it as noise and simply discard it.

Before putting sub-events on a timeline, we perform a final check to see whether they have not been validated in the previous time window. If yes, it means that an action overlaps two time-windows, and the timestamp of the event must be updated, matching the time of the first occurrence. We consider two events identical if: *i)* they mention the same action and participants; *ii)* the number of tweets reporting the more recent action is lower than the number of tweets on the old one.

4 Experiments

This section reports on the experiments we carried out to evaluate the proposed framework. We first present the dataset, then we describe the experimental setting and we discuss the obtained results.

4.1 Dataset

We experiment our framework on the Hackatal 2016 dataset[3], collected during the EURO 2016 Championship. A set of keywords were manually defined, including hashtags (#euro, #euro2016, #football) and the names of the teams involved in the competition (e.g. France) as well as their short names (e.g. #FRA) and hashtags related to current games (e.g. #FRAROM for the game between France and Romania). For each game, tweets were collected for a two-hour time span, starting at the beginning of the game. For comparisons and to limit the complexity of the processing pipeline, we limit our analysis to tweets in English.

The dataset also contains the summary of the salient sub-events in each game, retrieved from journalistic reports (e.g. LeFigaro[4]). We consider these summaries as the ground truth while evaluating our approach. These summaries are defined as a set of triples ⟨`time, action, participant`⟩ where "time" is the time the sub-event occurs, the "action" is the type of the sub-event and "participants" are players or teams involved in the action. The sub-events include: the beginning of the periods (F1P, D1P), end of the periods (F1P, D2P), Shoot (TIR), Goal (BUT), Substitution (CGT), Red card (CRO) and Yellow card (CJA). A few examples of such sub-events in a match are reported in Table 2.

Algorithm 1 Algorithm to process a given event-graph to retrieve important sub-events.

1: **function** GRAPH_PROCESSING(G_t, G_{t-1}, t) ▷ G_t - Event graph at time t, G_{t-1} - Event graph at t-1, t - current time
2: $G =$ **merge** (G_t, G_{t-1})
3: $E = \emptyset$
4: **for** $vertex \in G.vertices()$ **do**
5: **if** $vertex is foot\_action$ **then**
6: $P = G.neighbors(node)$
7: $a = node.action$
8: $\varphi_{a,t} =$ **compute** (a, t) ▷ equation 1
9: $z_{a,t} =$ **compute** (a, P, t) ▷ equation 3
10: **for** $z \in z_{a,t}$ **do**
11: **if** $z \geq \varphi_{a,t}$ **then**
12: $event = (a, p, t)$
13: E **append** (a, p, t)
14: G **delete** (a, p)
15: **end if**
16: **end for**
17: **end if**
18: **end for**
19: **end function**

4.2 Experimental Setting

We simulate the Twitter stream by grouping the tweets related to a game in intervals of two minutes, which we refer to as *time-windows*. Thus, we collect all the tweets published in a time-window in a single document which we give in input to our algorithm. In the preprocessing phase, we remove re-tweets if the original tweet is already in the col-

[3]`http://hackatal.github.io/2016/`.

[4]`http://sport24.lefigaro.fr`

Time	Action	Participants
15:02	D1P	–
15:09	TIR	Sterling
...	...	...
15:44	BUT	Bale
15:48	F1P	–
16:04	CGT	Sterling;Vardy
16:18	BUT	Vardy

Table 2: A few examples of the sub-events that occurred in the game between England and Wales.

lection, and we consider one tweet per user in a time window. The input tweets are then analyzed with GATE. We use the JGraph library (Naveh et al., 2008) to create the event-graph. At each time-window, we create a new graph to model the relation between actions and participants detected in tweets. We process the event-graph with Algorithm 1 to detect real sub-events found in tweets.

4.3 Evaluation Strategies

We report on two different evaluation strategies. In the first one, we compare the output of our framework against the state of the art approach described in (Alonso and Shiells, 2013). There, the authors detect sub-events by identifying spikes in the Twitter stream. Since they do not detect participants, in this first comparison we also limit our evaluation to the action timeline, letting out additional information. We also compare the results with the gold standard timeline from manually created summaries by sports journalists. We show the results through a graphical representation for three sample matches (Figures 4, 5 and 6).

In the second evaluation strategy, we evaluate our approach against the gold standard data (see above) in term of precision, recall and f-measure. This time we include also the sub-event type, the time and participants information. We adopt three evaluation strategies, namely *complete* matching, *partial* matching and *loose* matching. In the complete matching mode, we evaluate each sub-event detected by our system by taking into account the type of the sub-event, the participants and the time. A sub-event is considered correct if all three elements are correctly identified. In the partial mode, we consider the time and the type of the sub-events; and in the loose mode, we only consider the type. We set the error margin to 2 minutes while comparing the time, since this is the duration of the time-windows used to build the temporal graphs. We report P/R/F1 for the same sample matches described above, as well as an average of the scores for 24 matches in the first stage of the competition (Table 3).

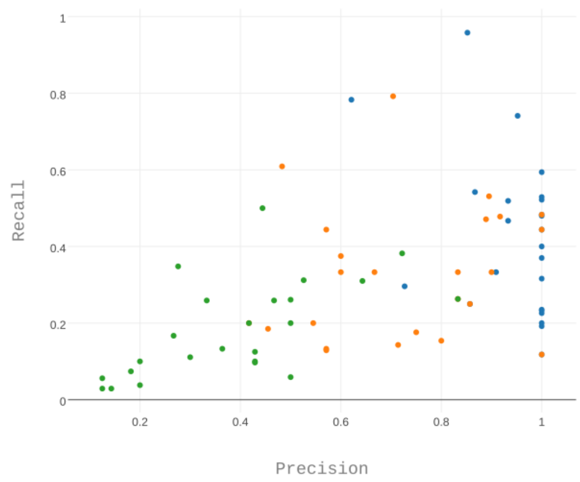

Figure 3: Precision Recall chart of the performances of our approach. X-axis is the average precision and Y-axis the average recall. Blue dots represent the loose matching, orange dots the partial matching and green dots the complete matching.

4.4 Results and discussion

The overall evaluation concerning the first 24 games in the EURO 2016 Championship (Table 3) shows that the approach is very accurate in some cases, while it suffers from low performance, especially recall, in other settings. If we compare the different actions (left-most columns in the table), we observe that the best performance is obtained when recognizing the start and the end of the match (last line in the table). For other actions, the performance varies across the three evaluation modes. For example, when considering participants to *shoot* actions, the approach fails to identify the correct player, probably because other players such as the defender and the goalkeeper are likely to be mentioned in the same tweet. In Figure 3 we provide a global overview of Precision and Recall obtained on the whole dataset with the different evaluation strategies, with each dot corresponding to a match.

We further focus on three sample matches, which were selected to compare our approach with (Alonso and Shiells, 2013). We plot in Figures 4, 5 and 6 the sub-events detected by (Alonso and Shiells, 2013), those detected by our approach as well as those present in the gold standard. We also report in Tables 4, 5 and 6 P/R/F1 measures ac-

	Loose			Partial			Complete		
actions	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
goal	0.745	0.512	0.549	0.670	0.456	0.493	0.623	0.405	0.444
card	0.758	0.560	0.622	0.693	0.506	0.568	0.600	0.433	0.516
subt	0.859	0.629	0.693	0.627	0.460	0.510	0.501	0.374	0.438
shoot	0.643	0.203	0.292	0.571	0.185	0.264	0.548	0.167	0.243
period	0.814	0.656	0.706	0.655	0.517	0.562	0.585	0.462	0.523

Table 3: Experimental results of our approach for 24 games in the first stage of the Euro 2016 dataset

cording to the loose, partial and complete evaluation strategy.

The first game considered was played between England and Wales and gained particular attention on Twitter. Figure 4 shows the distribution of tweets during the game (in gray), distinguishing between tweets explicitly mentioning England (red line) and Wales (green). The blue dots correspond to the sub-events identified by (Alonso and Shiells, 2013)'s approach, while those detected by our approach and the ground truth are represented with yellow and green dots, respectively. The graphical representation shows that there is a significant correspondence between the sub-events detected by our approach and the gold standard ones. We can also observe that (Alonso and Shiells, 2013) fails to detect sub-events that do not produce spikes in the volume of tweets (e.g. shoots).

Table 4 shows for the same match the average performance (P/R/F1) of our approach compared to the ground truth. In this case, our performance is affected by problems in detecting actions of type *substitution* and *shoots* (tweets mostly contain complains by England fans against *Kane* and *Sterling* who seemed to have missed a lot of opportunities to score for England in the first period).

Methods	Prec	Rec	F-score
loose	0.852	0.958	0.902
partial	0.630	0.708	0.667
complete	0.444	0.500	0.470

Table 4: Evaluation performance for the game between England and Wales.

A second example is the match between France and Romania, represented in Figure 5. Although the game was quite debated on Twitter, a few spikes were detected in the stream. In fact, during the first period the teams were barely mentioned, as indicated by the red and green curves on the graph. Instead, other teams were mentioned, which were not directly involved in the game. The second period seemed to be more interesting in

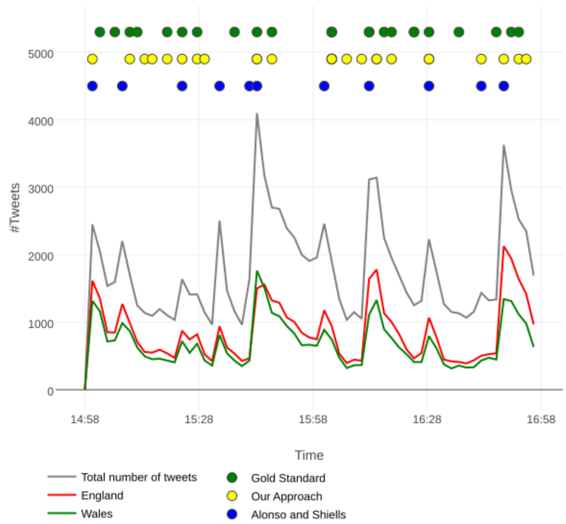

Figure 4: Sub-events for the game England vs Wales.

terms of sub-events. In Table 5, we show the performance of our approach on this game. We obtain a 91.3% precision in the loose mode, since we detect 23 out of 34 sub-events in the game compared to 9 identified by (Alonso and Shiells, 2013), and 21 of the detected sub-events were associated to the correct actions. However, the latency between the sub-events detected by our approach compared to the ground truth contributes in decreasing the performance of our approach in both intermediate and complete matching. For example, there is a huge peak at time 22:24 when the player *Stancu* equalizes for Romania, but we detect this action four minutes later since most of the tweets in that time span discuss the penalty issue rather than the goal. Many sub-events in the game, mostly actions by Romania, were not mentioned in any tweet in the dataset. For example, no tweets mentioned the shoot by Pintilii at time 21:04.

As a third example, we consider the game between Belgium and Italy, that was less popular in terms of tweets than the ones described so far. A few peaks are detected in the game, as shown in Figure 6. This affects negatively the number of

Figure 5: Sub-events for the game France vs Romania.

Figure 6: Sub-events for the game Belgium vs Italy.

Methods	Prec	Rec	F-score
loose	0.913	0.656	0.763
partial	0.696	0.500	0.582
complete	0.609	0.438	0.510

Table 5: Evaluation performance for the game between France and Romania.

sub-events found by (Alonso and Shiells, 2013), while our approach proves to have a better coverage, even if recall is on average lower than for the other matches. In most cases, we detect mentions of the actions, but we fail to detect the participants. Table 6 shows the overall performance of our approach. In the ground truth there were only a few tweets related to this game, and $\sim$ 50% of them were shoots. Our approach failed to identify such events, impacting on the recall. On the other hand, all the events detected were correct, accounting for 100% precision in the loose mode, and $\sim$ 85% in the complete mode.

Methods	Prec	Rec	F-score
loose	1.000	0.448	0.619
partial	0.923	0.414	0.572
complete	0.846	0.379	0.523

Table 6: Evaluation performance for the game between Belgium and Italy.

5 Conclusion and Future Work

In this paper, we have described a framework to generate timelines of salient sub-events in sports games exploiting information contained in tweets. We use the GATE system enriched with information provided by domain knowledge bases to detect mentions of actions and participants, as well as their relations in the sports domain (e.g. players and teams). Exploiting the self-contained nature of tweets, we made the hypothesis that entities that appear in the same tweets can be considered as related. We model the relationships between the entities in a temporal graph and use adaptive thresholds to measure the veracity of actions reported in tweets. Experiments on a dataset of tweets collected during the EURO 2016 Championship proved that our approach is able to accurately detect sub-events in sports games when compared to news on the same events reported by sports media. While previous approaches focused only on detecting the type of the most important sub-events, we extract and model a richer set of information, including almost every type of sub-event and participants involved in the actions.

In the future, we plan to extend our approach to cover other sports such as American football and basketball. To this end, we will extend our rules to detect relations between action and participants according to the rules that govern the games. Then, we will configure our framework to collect data from knowledge bases that provide information for these sports categories.

References

Omar Alonso and Kyle Shiells. 2013. Timelines as summaries of popular scheduled events. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, pages 1037–1044.

International Press Telecommunications Council. 2017. SportsML: A solution for sharing sports data. https://iptc.org/standards/sportsml-g2/. [Accessed 03-01-2017].

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Wenwen Dou, K Wang, William Ribarsky, and Michelle Zhou. 2012. Event detection in social media data. In *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*. pages 971–980.

C. Fellbaum. 1998. *WordNet. An Electronic Lexical Database*. MIT Press.

Chris Gibbs and Richard Haynes. 2013. A phenomenological investigation into how twitter has changed the nature of sport media relations. *International Journal of Sport Communication* 6(4):394–408.

John Hannon, Kevin McCarthy, James Lynch, and Barry Smyth. 2011. Personalized and automatic social summarization of events in video. In *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, pages 335–338.

Said Jai-Andaloussi, Imane El Mourabit, Nabil Madrane, Samia Benabdellah Chaouni, and Abderrahim Sekkaki. 2015. Soccer events summarization by using sentiment analysis. In *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, pages 398–403.

Erwin Kreyszig. 2007. *Advanced engineering mathematics*. John Wiley & Sons.

Mitsumasa Kubo, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2013. Generating live sports updates from twitter by finding good reporters. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01*. IEEE Computer Society, pages 527–534.

Markus Löchtefeld, Christian Jäckel, and Antonio Krüger. 2015. Twitsoccer: knowledge-based crowdsourcing of live soccer events. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*. ACM, pages 148–151.

Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. 2011. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pages 227–236.

Barak Naveh et al. 2008. Jgrapht. *Internet: http://jgrapht.sourceforge.net* .

Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. 2012. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, pages 189–198.

Selami Özsoy. 2011. Use of new media by turkish fans in sport communication: Facebook and twitter. *Journal of Human Kinetics* 28:165–176.

Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. 2010. Summarizing microblogs automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 685–688.

Cees GM Snoek and Marcel Worring. 2005. Multimedia event-based video indexing using time intervals. *IEEE Transactions on Multimedia* 7(4):638–647.

Cees GM Snoek, Marcel Worring, et al. 2003. Time interval based modelling and classification of events in soccer video. In *Proceedings of the 9th annual conference of the advanced school for computing and imaging (ASCI), Heijen*. Citeseer.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, pages 75–80.

Wei Xu, Ralph Grishman, Adam Meyers, and Alan Ritter. 2013. A preliminary study of tweet summarization using information extraction. *NAACL 2013* page 20.